

# Distributed Real-Time Embedded Video Processing

*Tiehan Lv*

*Wayne Wolf*

Dept. of EE, Princeton University

Phone: (609) 258-1424

Fax: (609) 258-3745

Email: [wolf@princeton.edu](mailto:wolf@princeton.edu)

*Burak Ozer*

Verificon Corp.

**Abstract:** The embedded systems group at Princeton University is building a distributed system for real-time analysis of video from multiple cameras. Most work in multiple-camera video systems relies on centralized processing. However, performing video computations at a central server has several disadvantages: it introduces latency that reduces the response time of the video system; it increases the amount of buffer memory required; and it consumes network bandwidth. These problems cause centralized video processing systems to not only provide lower performance but to use excess power as well. A deployable multi-camera video system must perform distributed computation, including computation near the camera as well as remote computations, in order to meet performance and power requirements.

Smart cameras combine sensing and computation to perform real-time image and video analysis. A smart camera can be used for many applications, including face recognition and tracking. We have developed a smart camera system [Wol02] that performs real-time gesture recognition. This system, which currently runs on a Trimedia TM-100 VLIW processor, classifies gestures such as *walking*, *standing*, *waving arms*. It currently runs at 25 frames/sec on the Trimedia processor. The application uses a number of standard vision algorithms as well as some improvements of our own; the details of the algorithms are not critical to the distributed system research we propose here.

However, real-time vision is very well suited to distributed system implementation. Using multiple cameras simplifies some important problems in video analysis. Occlusion causes many problems in vision; for example, when the subject turns such that only one arm can be seen from a single camera, the algorithms must infer that the arm exists in order to confirm that the subject in front of the camera is a person and not something else. When views are available from multiple cameras, the data can be fused to provide a global view of the subject that provides more complete information for higher-level analysis. Multiple cameras also allow us to replace mechanical panning and zooming with electronic panning and zooming. Electronically panned/zoomed cameras do not have inertia that affect tracking; they are also more reliable under harsh environmental conditions.

Report Documentation Page				Form Approved OMB No. 0704-0188	
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.					
1. REPORT DATE <b>20 AUG 2004</b>		2. REPORT TYPE <b>N/A</b>		3. DATES COVERED <b>-</b>	
4. TITLE AND SUBTITLE <b>Distributed Real-Time Embedded Video Processing</b>				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S)				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) <b>Dept. of EE, Princeton University; Verificon Corp.</b>				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT <b>Approved for public release, distribution unlimited</b>					
13. SUPPLEMENTARY NOTES <b>See also ADM001694, HPEC-6-Vol 1 ESC-TR-2003-081; High Performance Embedded Computing (HPEC) Workshop (7th)., The original document contains color images.</b>					
14. ABSTRACT					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT <b>UU</b>	18. NUMBER OF PAGES <b>15</b>	19a. NAME OF RESPONSIBLE PERSON
a. REPORT <b>unclassified</b>	b. ABSTRACT <b>unclassified</b>	c. THIS PAGE <b>unclassified</b>			

Processing in the distributed smart camera is inherently distributed. Sending raw frames to a central node for processing would consume enormous amounts of bandwidth; that bandwidth not only increases the cost of the network, it consumes considerable energy as well. Performing low-level processing at each camera, then sending abstractions of the frames to other nodes that can combine the results from several frames saves considerable power. Distributed processing is also probably the most effective approach to meeting real-time deadlines and low latency. The smart camera system will often be used as input to a decision-making system or person, so low latency is an important criteria. Using multiple nodes to process the data should considerably reduce the latency in obtaining the result.

We believe that a combination of design-time and run-time decisions are required for the successful deployment of video processing networks. Some run-time decisions are necessary because some characteristics of the system will not be known until run time, and those characteristics may change during operation of the network. Furthermore, the very configuration of the network may not be known at design time, as it may depend on the size of the area to be monitored, the physical constraints of installing nodes, etc. The overall hardware and software architecture of the network must be designed to operate well not just at a single design point, but across a range of possible configurations and operating decisions.

We are developing a middleware-based system for video analysis. The DVM (distributed video middleware) runs on top of the operating system (Linux) on each node. The DVM layer deals with video objects. The video objects may be described in any of several ways: regions of an image, curves that represent sections of the image, etc. The video objects represent the current state of analysis. The DVM layer manages the location of the video object data as it goes through the various processing stages. The DVM may be guided by design-time information provided by tools, such as the relative amounts of data and processing time required for various operations; it also makes use of run-time information that helps it make the best use of available resources.

A DVM-based architecture leverages COTS operating systems. It also helps to minimize the amount of work that is required to port a batch-oriented video application to a distributed platform. Relatively few video processing algorithm experts are also adept at distributed computing. The video object model allows them to map their video algorithms onto data structures and let the DVM layer manage the allocation and scheduling tasks required to run in distributed mode.

## References

[Wol02] Wayne Wolf, Burak Ozer, and Tichan Lv, "Smart cameras as embedded systems," IEEE Computer, 35(9) September 2002, pp. 48-53.

# Distributed Real-Time Embedded Video Processing

*Tiehan Lv (1), Burak Ozer (2), Wayne Wolf (1)*

*(1) Dept. of EE, Princeton University*

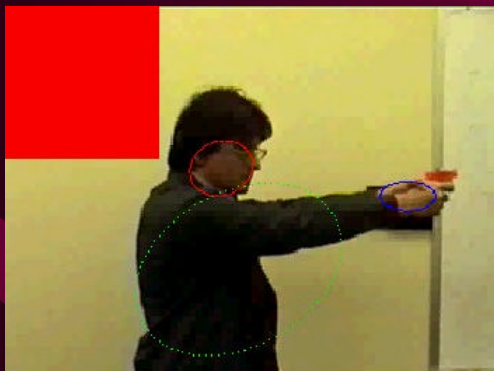
*(2) Verificon Corp.*

# Smart Camera Systems

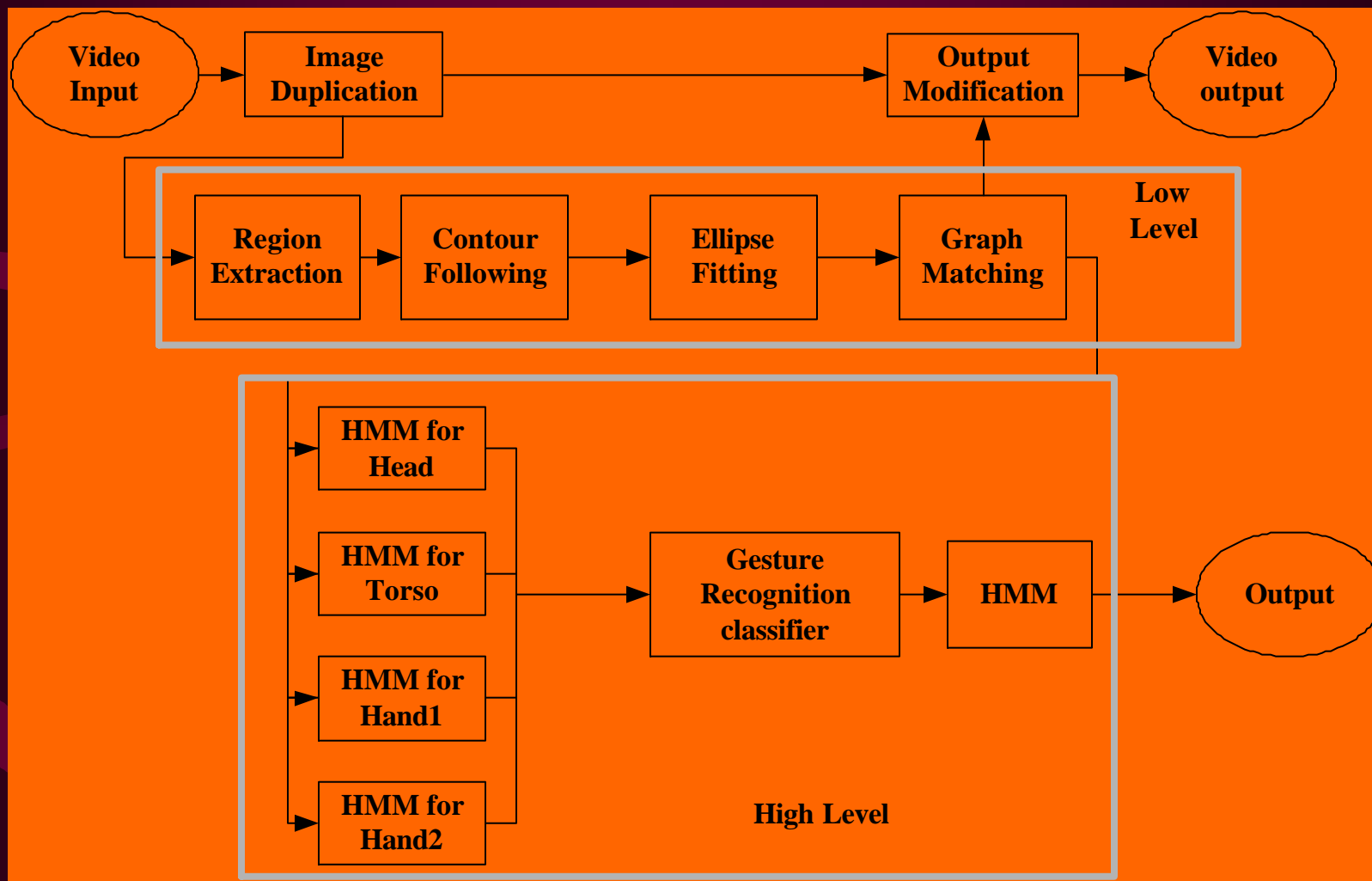
- A smart camera is a video surveillance system that is able to identify body parts and objects and then recognize the activity of people or objects in the scene.



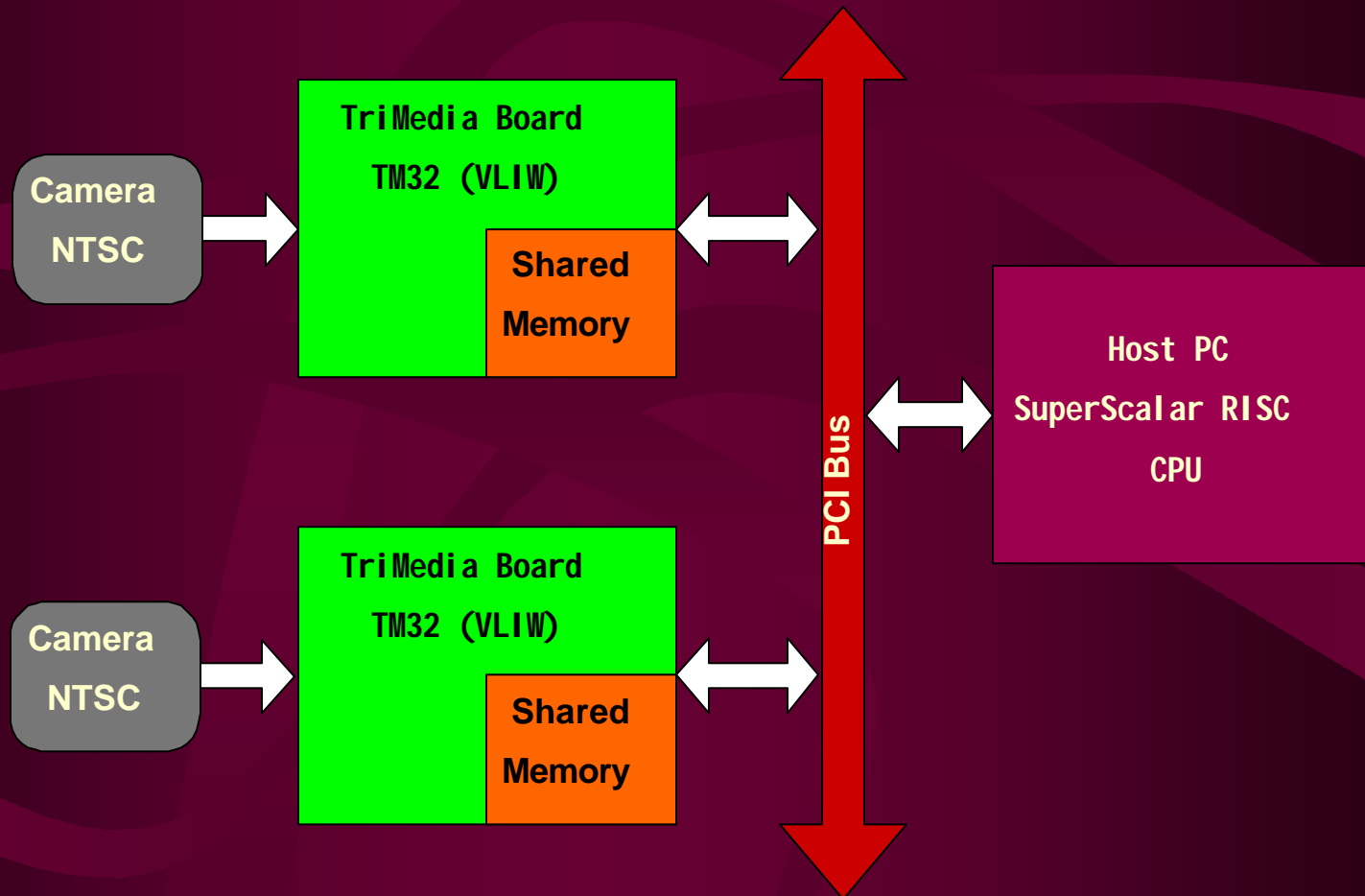
# Sample Scenarios



# Algorithms



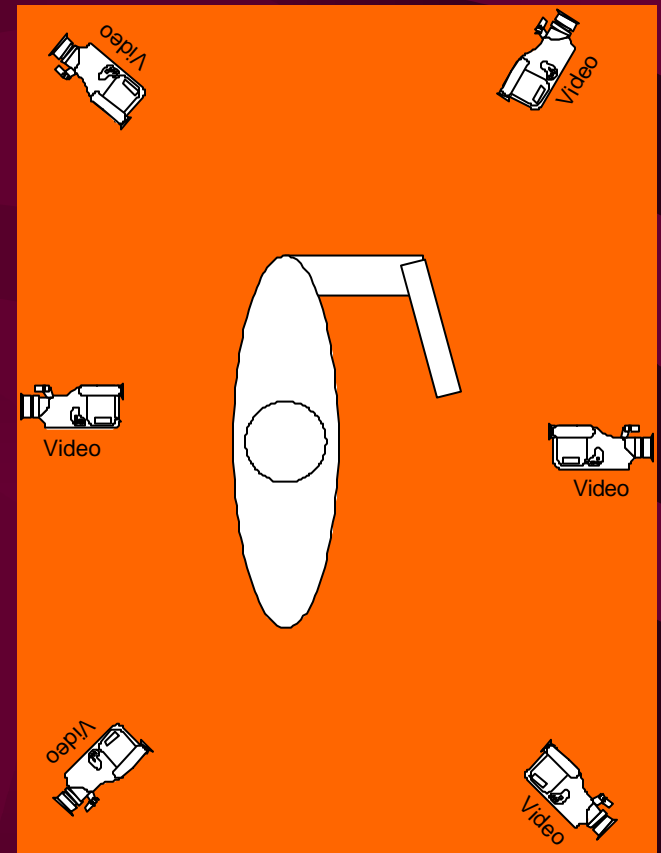
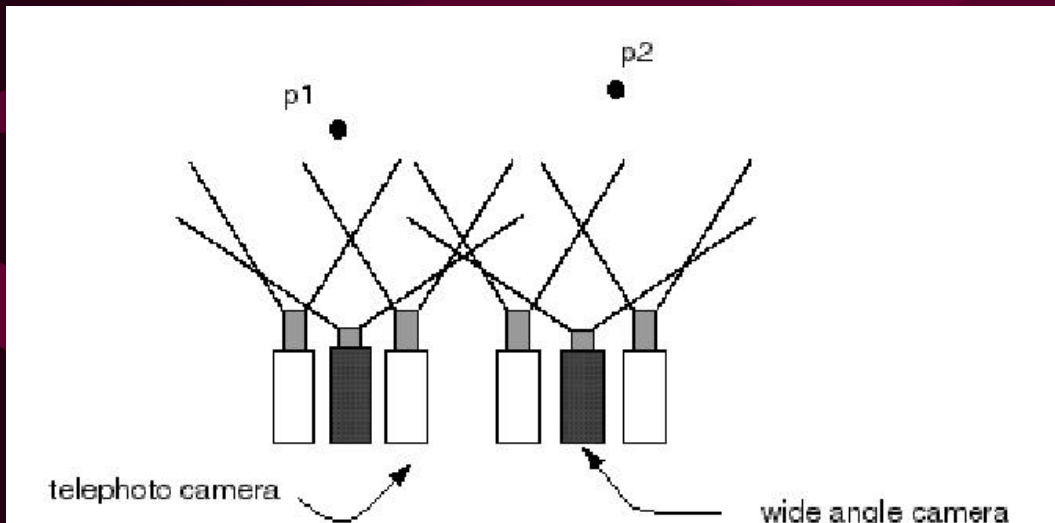
# Architecture of a Smart Camera System





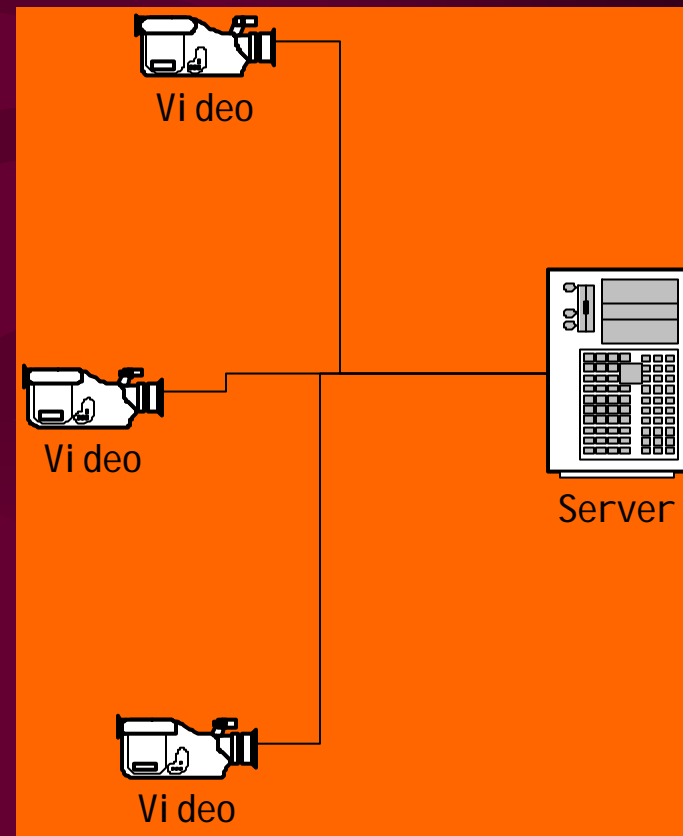
# Multiple Camera Systems

- Electronically Panning&Zooming
- Occlusion



# Centralized Processing

- Storage Cost
- Latency
- Communication Load
- Power



# Centralized Processing vs. Distributed Processing

- Raw Data vs. Abstract Representation
  - Network Load
  - Energy
- Latency
- Processing Power

# Design Time Decisions and Run-time Decisions

- Configuration
  - Processor
  - Special Functional Units
  - Hardware Architecture
  - Operating System
- Efficiency
- Flexibility

# Distributed Video Middleware

- The Concept of Layers
- Trans-platform Development
- Trans-platform Communication

# Distributed Video Middleware

Video Processing Application

DVM

Operating Systems

# Distributed Video Middleware

- Separate video processing algorithm and operating system
- Algorithm researcher focus on video processing
- Facilitate porting application to different systems

# Conclusion and Future Work

- Distributed smart camera systems have advantages over traditional centralized processing systems
- Design time decisions and run-time decisions need to be combined to form an optimal solution
- Distributed video middleware can facilitate research and application development